

FILEID**DATECN

D 6

DDDDDDDD	AAAAAA	TTTTTTTT	EEEEEEEEE	CCCCCCC	NN	NN
DDDDDDDD	AAAAAA	TTTTTTTT	EEEEEEEEE	CCCCCCC	NN	NN
DD DD	AA AA	TT TT	EE	CC	NN	NN
DD DD	AA AA	TT TT	EE	CC	NNNN	NN
DD DD	AA AA	TT TT	EE	CC	NNNN	NN
DD DD	AA AA	TT TT	EE	CC	NN NN	NN
DD DD	AA AA	TT TT	EE	CC	NN NN	NN
DD DD	AAAAAA	TT	EE	CC	NN NNNN	NN
DD DD	AAAAAA	TT	EE	CC	NN NNNN	NN
DD DD	AA AA	TT	EE	CC	NN NN	NN
DD DD	AA AA	TT	EE	CC	NN NN	NN
DDDDDDDD	AA AA	TT	EEEEEEEEE	CCCCCCC	NN	NN
DDDDDDDD	AA AA	TT	EEEEEEEEE	CCCCCCC	NN	NN

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

1 0001 0
2 0002 0 MODULE DATECN (LANGUAGE (BLISS32) .
3 0003 0 IDENT = 'V04-000'
4 0004 0) =
5 0005 1 BEGIN
6 0006 1 !*****
7 0007 1 !*****
8 0008 1 !*
9 0009 1 !* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 !* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 !* ALL RIGHTS RESERVED.
12 0012 1 !*
13 0013 1 !* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 !* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 !* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 !* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 !* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 !* TRANSFERRED.
19 0019 1 !*
20 0020 1 !* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 !* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 !* CORPORATION.
23 0023 1 !*
24 0024 1 !* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 !* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 !*
27 0027 1 !*
28 0028 1 !*****
29 0029 1 !**
30 0030 1 !**
31 0031 1 !**
32 0032 1 !* FACILITY: MTAACP
33 0033 1 !*
34 0034 1 !* ABSTRACT:
35 0035 1 !* CONVDATE_R2J converts dates from DD-MMM-YY format to (blank)YYDDD
36 0036 1 !* CONVDATE_J2R converts dates from (blank)YYDDD to DD-MMM-YY
37 0037 1 !*
38 0038 1 !*
39 0039 1 !* ENVIRONMENT:
40 0040 1 !*
41 0041 1 !* STARLET operating system, including privileged system services
42 0042 1 !* and internal exec routines.
43 0043 1 !*
44 0044 1 !--
45 0045 1 !*
46 0046 1 !*
47 0047 1 !*
48 0048 1 !* AUTHOR: D. H. GILLESPIE, CREATION DATE: 31-MAY-77 9:00
49 0049 1 !*
50 0050 1 !* MODIFIED BY:
51 0051 1 !*
52 0052 1 !* V02-005 DMW00055 David Michael Walp 30-Nov-1981
53 0053 1 !* Fixed problem of 366 days in a none leap year when converting
54 0054 1 !* from Julian to VMS
55 0055 1 !*
56 0056 1 !* V02-004 DMW00039 David Michael Walp 2-Oct-1981
57 0057 1 !* Handle Julian Date "00000" and Regular Dates less than 1900

```
58 0058 1
59 0059 1
60 0060 1
61 0061 1 V02-003 REFORMAT Maria del C. Nasr 30-Jun-1980
62 0062 1 A0002 MCN0012 Maria del C. Nasr 15-Feb-1980 3:20
63 0063 1 Check for date out of range when converting to either Julian
64 0064 1 format or regular date
65 0065 1 !**
66 0066 1
67 0067 1 LIBRARY 'SY$LIBRARY:LIB.L32';
68 0068 1
69 0069 1 REQUIRE 'SRC$:MTADEF.B32';
70 0453 1
71 0454 1 FORWARD ROUTINE
72 0455 1 CONVDATE_R2J, : regular to julian
73 0456 1 CONVDATE_J2R; : julian to regular
74 0457 1
75 0458 1 EXTERNAL ROUTINE
76 0459 1 LIB$CVT_DTB : ADDRESSING_MODE ( ABSOLUTE ), : convert decimal to binary
77 0460 1 SY$FAO : ADDRESSING_MODE ( ABSOLUTE ); : formatted ascii output
78 0461 1
79 0462 1 GLOBAL BIND
80 0463 1 ZERO_JDATE = UPLIT ( ' 00000' );
81 0464 1
82 0465 1 BIND
83 0466 1 DAYTBL = UPLIT WORD ( 0,31,59,90,120,151,181,212,243,273,304,334,365 )
84 0467 1 : VECTOR [,WORD],
85 0468 1 MONTBL = UPLIT BYTE ( 'JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC' )
86 0469 1 : VECTOR [CH$ALLOCATION(48)];
```

```
88 0470 1 GLOBAL ROUTINE CONVDATE_R2J ( REGDATE, ANSIJDATE ) =
89 0471 1
90 0472 1 ++
91 0473 1
92 0474 1 : FUNCTIONAL DESCRIPTION:
93 0475 1 : convert from DD-MMM-YYYY to (blank)YYDDD
94 0476 1
95 0477 1
96 0478 1 : CALLING SEQUENCE:
97 0479 1 : CONVDATE_R2J ( ARG1, ARG2 )
98 0480 1
99 0481 1 : INPUT PARAMETERS:
100 0482 1 : ARG1 address of input string DD-MMM-YYYY
101 0483 1 : ARG2 address of output string (blank)YYDDD
102 0484 1
103 0485 1 : IMPLICIT INPUTS:
104 0486 1 : none
105 0487 1
106 0488 1 : OUTPUT PARAMETERS:
107 0489 1 : output string pointed to by ARG2
108 0490 1
109 0491 1 : IMPLICIT OUTPUTS:
110 0492 1 : none
111 0493 1
112 0494 1 : ROUTINE VALUE:
113 0495 1 : 1 if successful
114 0496 1 : 0 syntax error
115 0497 1
116 0498 1 : SIDE EFFECTS:
117 0499 1 : none
118 0500 1
119 0501 1 :--
120 0502 1
121 0503 2 BEGIN
122 0504 2
123 0505 2 LOCAL
124 0506 2 : CONVOUTPUT : VECTOR [2], : output descriptor
125 0507 2 : MONTH, : index into month table
126 0508 2 : DAYS, : number of days
127 0509 2 : REGADR, : char ptr to reg date string
128 0510 2 : YEAR, : number of year
129 0511 2 : LPYEAR; : leap year
130 0512 2
131 0513 2 BIND
132 0514 2 : BASE_YEAR = UPLIT BYTE ( '19' ),
133 0515 2 : ZERO_YEAR = UPLIT BYTE ( '0000' ),
134 0516 2 : CONVCONTROL = DESCRIPTOR ( ' !AD!3ZL' ),
135 0517 2 : BYEAR = YEAR : BLOCK [1];
136 0518 2
137 0519 2 : setup addr pointer and convert DD to binary # of days
138 0520 2
139 0521 2 IF (.REGDATE)<0,8> EQL '
140 0522 2 THEN
141 0523 3 BEGIN
142 0524 3 : REGADR = .REGDATE + 1;
143 0525 3 : IF NOT LIBSCVT_DTB ( 1, .REGADR, DAYS ) THEN RETURN 0;
144 0526 3 END
```

```
145      0527 2      ELSE
146      0528 2      IF NOT LIBSCVT_DTB ( 2, .REGDATE, DAYS ) THEN RETURN 0;
147      0529 2
148      0530 2      | point into date to MMM
149      0531 2
150      0532 2      REGADR = .REGDATE + 3;
151      0533 2
152      0534 2      | lookup MMM in table to discover # of days in year preceding current month
153      0535 2
154      0536 2      MONTH = 0;
155      0537 2      WHILE 1
156      0538 2      DO
157      0539 3      BEGIN
158      0540 3      IF CH$EQ(L ( 3, .REGADR, 3, MONTBL.MONTH ), 0 ) THEN EXITLOOP;
159      0541 3      MONTH = .MONTH + 1;
160      0542 3      IF .MONTH GEQU 12 THEN RETURN 0;
161      0543 2      END;
162      0544 2
163      0545 2      | convert year
164      0546 2
165      0547 2      REGADR = .REGDATE + 7;
166      0548 2      IF NOT LIBSCVT_DTB ( 4, .REGADR, YEAR ) THEN RETURN 0;
167      0549 2
168      0550 2      | determine if year is leap year
169      0551 2
170      0552 2      LPYEAR = 0;
171      0553 2      IF .BYEAR [ 0, 0, 2, 0 ] EQLU 0 THEN LPYEAR = 1;
172      0554 2
173      0555 2      | decide if legal day in month
174      0556 2
175      0557 3      IF ( .DAYS GTRU ( .DAYTBL.MONTH + 1 ) - .DAYTBL.MONTH ) )
176      0558 3      AND ( NOT ( .LPYEAR AND ( .MONTH EQLU 1 ) AND ( .DAYS EQLU 29 ) ) )
177      0559 2      THEN RETURN 0;
178      0560 2
179      0561 2      | calculate the days into the year
180      0562 2
181      0563 2      DAYS = .DAYS + .DAYTBL.MONTH;
182      0564 2      IF .MONTH GTRU 1 THEN DAYS = .DAYS + .LPYEAR;
183      0565 2
184      0566 2      | if the year is less than 1900, then julian date must be zero
185      0567 2
186      0568 2      IF CH$LSS ( 2, .REGADR, 2, BASE_YEAR )
187      0569 2      THEN
188      0570 3      BEGIN
189      0571 3      CH$MOVE ( 4, ZERO_YEAR, .REGADR );
190      0572 3      DAYS = 0;
191      0573 2      END;
192      0574 2
193      0575 2      | setup output string addr and length
194      0576 2
195      0577 2      CONVOUTPUT[0] = 6;
196      0578 2      CONVOUTPUT[1] = .ANSIJDATE;
197      0579 2
198      0580 2      | setup yy char ptr
199      0581 2
200      0582 2      REGADR = .REGDATE + 9;
201      0583 2
```

```
:
: 202 0584 2      ! format output string
: 203 0585 2
: 204 0586 2      SYSSFAO( CONVCONTROL, 0, CONVOUTPUT, 2, .REGADR, .DAYS );
: 205 0587 2      RETURN 1;
: 206 0588 2
: 207 0589 1      END;                                ! end routine
```

```
.TITLE DATECN
.IDENT \V04-000\
.PSECT $CODES,NOWRT,2

0111 00F3 00D4 0085 0097 00 00 30 30 30 30 30 30 20 00000 P.AAA: .ASCII \ 00000\<0><0>
52 50 41 20 52 41 4D 20 42 45 46 20 4E 41 4A 00008 P.AAB: .WORD 0, 31, 59, 90, 120, 151, 181, 212, 243, -
55 41 20 4C 55 4A 20 4E 55 4A 20 59 41 4D 20 0001C P.AAC: .ASCII \JAN FEB MAR APR MAY JUN JUL AUG SEP OCT \
20 54 43 4F 20 50 45 53 20 47 00031
20 43 45 44 20 56 4F 4E 00040
39 31 00042 P.AAD: .ASCII \NOV DEC \
40 5A 33 21 30 30 30 30 00052 P.AAE: .ASCII \19\
4C 5A 33 21 44 41 21 20 00054 P.AAF: .ASCII \0000\
00000008, 00060 P.AAG: .ASCII \!AD!3ZL\
00000000, 00064 P.AAF: .LONG 8
00000000, 00064 P.AAF: .ADDRESS P.AAG
```

```
ZERO_JDATE== P.AAA
DAYTBL= P.AAB
MONTHBL= P.AAC
BASE_YEAR= P.AAD
ZERO_YEAR= P.AAE
CONVCONTROL= P.AAF
.EXTRN LIB$CVT_DTB, SYSSFAO
```

58	00000000G	01FC 00000	.ENTRY CONVDATE R2J, Save R2,R3,R4,R5,R6,R7,R8	: 0470
57	94	9F 9E 00002	MOVAB @LIB\$CVT_DTB, R8	
5E	04	AF 9E 00009	MOVAB DAYTBL, R7	
56	04	10 C2 0000D	SUBL2 #16, SP	
20		AC D0 00010	MOVL REGDATE, R6	
		66 91 00014	CMPB (R6), #32	
		OC 12 00017	BNEQ 1\$	
54	01	A6 9E 00019	MOVAB 1(R6), REGADR	
	4010	8F BB 0001D	PUSHR #^M<R4,SP>	
		01 DD 00021	PUSHL #1	
		06 11 00023	BRB 2\$	
	4040	8F BB 00025	PUSHR #^M<R6,SP>	
		02 DD 00029	PUSHL #2	
68		03 FB 0002B	CALLS #3, LIB\$CVT_DTB	
27		50 E9 0002E	BLBC R0, 5\$	
54	03	A6 9E 00031	MOVAB 3(R6), REGADR	
		55 D4 00035	CLRL MONTH	
	1A A745	DF 00037	PUSHAL MONTBL[MONTH]	
9E	64	03 29 0003B	CMPC3 #3, (REGADR), a(SP)+	
		09 13 0003F	BEQL 4\$	
		55 D6 00041	INCL MONTH	
	0C	55 D1 00043	CMPL MONTH, #12	
		EF 1F 00046	BLSSU 3\$	

54	07	7C 11 00048	BRB	10\$		0547
	04	A6 9E 0004A	MOVAB	7(R6), REGADR		0548
		AE 9F 0004E	PUSHAB	YEAR		
		54 DD 00051	PUSHL	REGADR		
		04 DD 00053	PUSHL	#4		
68	03	FB 00055	CALLS	#3, LIB\$CVT_DTB		
68	50	E9 00058	BLBC	R0, 10\$		
03	04	51 D4 0005B	CLRL	LPYEAR		0552
		AE 93 0005D	BITB	BYEAR, #3		0553
		03 12 00061	BNEQ	6\$		
51	01	D0 00063	MOVL	#1, LPYEAR		
50	02	A745 3C 00066	MOVZWL	DAYTBL+2[MONTH], R0		0557
52		6745 3C 0006B	MOVZWL	DAYTBL[MONTH], R2		
50		52 C2 0006F	SUBL2	R2, R0		
50		6E D1 00072	CMPL	DAY\$, R0		
		0D 1B 00075	BLEQU	7\$		
4C	51	E9 00077	BLBC	LPYEAR, 10\$		0558
01	55	D1 0007A	CMPL	MONTH, #1		
	47	12 0007D	BNEQ	10\$		
1D	6E	D1 0007F	CMPL	DAY\$, #29		
		42 12 00082	BNEQ	10\$		
50	6745	3C 00084	MOVZWL	DAYTBL[MONTH], R0		0563
6E		50 C0 00088	ADDL2	R0, DAY\$		
01		55 D1 00C8B	CMPL	MONTH, #1		0564
		03 1B 0008E	BLEQU	8\$		
6E	51	C0 00090	ADDL2	LPYEAR, DAY\$		
64	02	29 00093	CMPC3	#2, (REGADR), BASE_YEAR		0568
	06	1E 00098	BGEQU	9\$		
64	4C	A7 D0 0009A	MOVL	ZERO_YEAR, (REGADR)		0571
		6E D4 0009E	CLRL	DAY\$		0572
08	AE	06 D0 000A0	MOVL	#6, CONVOUTPUT		0577
0C	AE	08 AC D0 000A4	MOVL	ANSIJDATE, CONVOUTPUT+4		0578
54	09	A6 9E 000A9	MOVAB	9(R6), REGADR		0582
		6E DD 000AD	PUSHL	DAY\$		0586
		54 DD 000AF	PUSHL	REGADR		
		02 DD 000B1	PUSHL	#2		
	14	AE 9F 000B3	PUSHAB	CONVOUTPUT		
		7E D4 000B6	CLRL	-(SP)		
00000000G	9F	58 A7 9F 000B8	PUSHAB	CONVCONTROL		
	50	06 FB 000BB	CALLS	#6, @SYSSFAO		
		01 D0 000C2	MOVL	#1, R0		0587
		04 000C5	RET			
		50 D4 000C6	CLRL	RO		0589
		10\$:	RET			

: Routine Size: 201 bytes. Routine Base: \$CODE\$ + 0068

```
0590 1 GLOBAL ROUTINE CONVDATE_J2R ( DATEREG, DATEANSIJ ) =  
0591 1  
0592 1 ++  
0593 1  
0594 1 FUNCTIONAL DESCRIPTION:  
0595 1 convert from (blank)YYDDD to DD-MMM-YYYY  
0596 1  
0597 1 CALLING SEQUENCE:  
0598 1 CONVDATE_J2R(ARG1,ARG2)  
0599 1  
0600 1 INPUT PARAMETERS:  
0601 1 ARG1 addr of output string DD-MMM-YYYY( 11 char )  
0602 1 ARG2 addr of input string (blank)YYDDD  
0603 1  
0604 1 IMPLICIT INPUTS:  
0605 1 none  
0606 1  
0607 1 OUTPUT PARAMETERS:  
0608 1 output string pointed to by ARG1  
0609 1  
0610 1 IMPLICIT OUTPUTS:  
0611 1 none  
0612 1  
0613 1 ROUTINE VALUE:  
0614 1 2 if date of " 00000"  
0615 1 1 if normal date  
0616 1 0 if an invalid date  
0617 1  
0618 1 SIDE EFFECTS:  
0619 1 none  
0620 1  
0621 1 --  
0622 1  
0623 2 BEGIN  
0624 2  
0625 2 LOCAL  
0626 2 CONVOUTPUT : VECTOR [2].  
0627 2 MONTH. : index into month table /3  
0628 2 DAYS. : # of days in month  
0629 2 YEAR. : # of years  
0630 2 JULADR. : addr of JULIAN char  
0631 2 LPYEAR. : 1 if leap year  
0632 2 FEB29: : 1 if FEB 29th  
0633 2  
0634 2  
0635 2 BIND CONVCONTROL = DESCRIPTOR ( '!2ZL-!AD-!4ZL' ),  
0636 2 BYEAR = YEAR : BLOCK [1];  
0637 2  
0638 2 ! if ANSI Julian date is zero, return with 2  
0639 2  
0640 2 IF CH$EQCL ( 6, .DATEANSIJ, 6, ZERO_JDATE ) THEN RETURN 2;  
0641 2  
0642 2 ! setup pointer to DDD in Julian input string  
0643 2  
0644 2 JULADR = CH$PLUS ( .DATEANSIJ, 3 );  
0645 2  
0646 2 ! convert to binary date
```

```
0647 2  IF NOT LIB$CVT_DTB ( 3, .JULADR, DAYS ) THEN RETURN 0;
0648 2
0649 2  ! backup pointer to yy and convert to binary and add century
0650 2
0651 2  JULADR = CH$PLUS ( .JULADR, -2 );
0652 2  IF NOT LIB$CVT_DTB ( 2, .JULADR, YEAR ) THEN RETURN 0;
0653 2  YEAR = .YEAR + 1900;
0654 2
0655 2  ! adjust days for leap year
0656 2  and make 366 of a non-leap year into a valid date
0657 2
0658 2
0659 2  LPYEAR = ( .BYEAR [ 0, 0, 2, 0 ] EQLU 0 );
0660 2  FEB29 = ( .DAYS EQLU 60 ) AND .LPYEAR;
0661 2  IF .LPYEAR
0662 3  THEN BEGIN
0663 3  IF .DAYS GTRU 59 THEN DAYS = .DAYS - 1;
0664 3  END
0665 2  ELSE IF .DAYS EQL 366
0666 2  THEN
0667 3  BEGIN
0668 3  YEAR = .YEAR + 1;
0669 3  DAYS = 1;
0670 2  END;
0671 2
0672 2  ! know days in year and if year is leap year, now is this a legal date
0673 2
0674 2  IF ( .DAYS EQLU 0 ) OR ( .DAYS GTRU 365 ) THEN RETURN 0;
0675 2
0676 2  ! find month
0677 2
0678 2  MONTH = 1;
0679 2  WHILE 1
0680 2  DO
0681 3  BEGIN
0682 3  IF ( .DAYS LEQU .DAYTBL[.MONTH] ) THEN EXITLOOP;
0683 3  MONTH = .MONTH + 1;
0684 3  IF .MONTH GEQU 13 THEN RETURN 0;
0685 2  END;
0686 2
0687 2  ! get date in month
0688 2
0689 2  MONTH = .MONTH - 1;
0690 2  DAYS = .DAYS - .DAYTBL[.MONTH] + .FEB29;
0691 2
0692 2  ! setup and format output string
0693 2
0694 2  CONVOUTPUT[0] = 11;
0695 2  CONVOUTPUT[1] = .DATEREG;
0696 2  SYS$FAO ( CONVCONTROL, 0, CONVOUTPUT, .DAYS, 3, MONTBL[.MONTH], .YEAR );
0697 2  RETURN 1;
0698 2
0699 1  END;                                ! routine
```

4C 5A 34 21 2D 44 41 21 2D 4C 5A 32 21 00131 P.AAI: .ASCII \!2ZL-!AD-!4ZL\

0000000D, 0013E P.AAH: .BLKB 2
00000000, 00140 .LONG 13
00000000, 00144 .ADDRESS P.AAI

CONVCONTROL= P.AAH

			36	11	0009D	7\$:	BRB	9\$					
			50	D7	0009F	8\$:	DECL	MONTH					
			6440	3C	000A1		MOVZWL	DAYTBL[MONTH], R3	0689				
			51	53	C2	000A5	SUBL2	R3, R1	0690				
			51	52	C1	000A8	ADDL3	FEB29, R1, DAYS					
			08	AE	0B	000AC	MOVL	#11, CONVOUTPUT	0694				
			0C	AE	04	000B0	MOVL	DATREG, CONVOUTPUT+4	0695				
					04	AE	PUSHL	YEAR	0696				
					1A	A440	PUSHAL	MONTBL[MONTH]					
						03	PUSHL	#3					
						0C	AE	DD	000BC				
						18	AE	DD	000BE				
						7E	9F	000C1	PUSHAB				
						C4	9F	000C4	CONVOUTPUT				
						0138	07	FB	000CA	CLRL	-(SP)		
							01	DO	000D1	PUSHAB	CONVCONTROL		
								04	000D4	CALLS	#7, @#SYS\$FAO		
								50	D4	000D5	9\$:	MOVL	#1, R0
									04	000D7	RET		0697
											CLRL	RO	0699
											RET		

: Routine Size: 216 bytes, Routine Base: \$CODE\$ + 0148

: 319 0700 1
: 320 0701 1 END
: 321 0702 1
: 322 0703 0 ELUDOM ! end module

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	544	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	4	0	1000	00:01.8

COMMAND QUALIFIERS

DATECN
VO4-000

B 7
16-Sep-1984 02:15:00 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:46:37 [MTAACP.SRC]DATECN.B32:1

Page 11
(3)

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:DATECN/OBJ=OBJ\$:DATECN MSRC\$:DATECN/UPDATE=(ENH\$:DATECN)

: 323 0704 0
: Size: 417 code + 127 data bytes
: Run Time: 00:11.4
: Elapsed Time: 00:35.0
: Lines/CPU Min: 3711
: Lexemes/CPU-Min: 16555
: Memory Used: 122 pages
: Compilation Complete

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

